CONCEPT FOR REAL-TIME LOCALIZATION BASED ON SMARTPHONE CAMERA AND IMU

Janek Stoeck, Harald Sternberg
Dept. Geodesy and Geoinformatics
HafenCity University Hamburg
Hamburg, Germany
firstname.name@hcu-hamburg.de

***Abstract*—This paper introduces a concept for a real-time localization system based on sensors available on smartphones. The proposed method relies on inertial measuring units, on one or more cameras and a given floor plan. Each of the used sensor produces position and orientation (pose) which will be fused with a kalman filter to get the best possible result. 3D point clouds generated from images of the camera are used to derive partly floor plans. By comparing the result to a given map the scaling and positioning fixes are adopted.**
Different possible platforms, like standard smartphones with one or smartphones with more than one camera, can be used to realize this concept. First experiments were done in two different approaches (with inertial and visual navigation) in order to get an idea what accuracies are achievable. The concept will be implemented and tested on a Samsung Galaxy S8.

***Index Terms*—indoor navigation, visual inertial navigation, smartphone, position estimation**

I. Introduction

Since 2011 the HafenCity University Hamburg (HCU) investigates smartphone sensors in context of indoor navigation. The utilization of smartphones for navigation, usually with the global navigation satellite system (GNSS) principle, is wide spread and commonly known, and the implemented sensors are a good base for developing an indoor navigation system. While GNSS is only available in outdoor environments, other solutions for the indoor environment has to be found. The purpose of this paper is to introduce a concept for indoor localization based on inertial and visual sensors, while the focus of this concept lies on the camera system.

II. Related Work

Nowadays cameras are well known for the purpose of visual odometry for robots, but they are also used in some pedestrian navigation systems. [1] and [2] using feature matching algorithms to determine the camera pose. [1] use a video stream to identify known key points, which are geo-referenced before, and inertial sensors. They propose a method of two phases: The first phase is an offline phase, where distinctive points (anchors) in the building are geo-referenced and processed with the Speeded Up Robust Features (SURF) algorithm, to construct a databese of reference images with known coordinates. In the second (online) phase, a smartphone's camera is used to take query pictures of these anchors, which will be send to a server to do a feature matching with the database images. The best fit is then used to fix the position, which is estimated by dead reckoning (DR) with step counter and heading of a magnetometer.
[2] propose a method to estimate a position using GNSS, inertial measuring unit (IMU) and the smartphone camera. The orientation is a solution of a bundle block adjustment with the input of all available sensor data.
Both of these methods use geo-referencing, either in runtime or before the online phase. The geo-referencing in an offline phase is a step, that may be done once, but it is time consuming and in the case of [1] heavily depends on the database. Furthermore, it requires a connection to some kind of server. The work of [2] depends on GNSS and as already mentioned this principle is not suitable for indoor navigation.
[3] developed a fusion algorithm for indoor navigation which uses inertial data and topological information to achieve an accuracy of less than 5 m in 70 %. His work is an inspiration for the following proposed method, because some parts of this work (like the step counter) are adopted from it. Further no external infrastructure is required and it is also developed and tested at the HCU.

III. Concept

The development of the following proposed concept aims to create a system, which is meant for indoor navigation and only relies on smartphone sensors. The developed algorithms should be executed in runtime on the smartphone. The first step is to initialize a start position. The navigation is realized by a DR based on IMU data and by image based navigation. Both approaches support each other in different ways. A given map of the building and a routing graph trough it are prerequisites for this method.

A. Proposed Method

Our proposed method can be split into two different modules, "inertial" and "visual" sensing. Fig. 1 shows the principle of the inertial sensing, which is realized by accelerometer, gyroscope and barometer. The accelerometer is used to implement a step counter like in [3]. The step counter is based on the accelerometer z-axis and has two conditions to be fulfilled. First, an initial maximal threshold has to be passed followed by passing the lower threshold. If these conditions are true, a step is detected. If the actual

https://doi.org/10.24355/dbbs.084-201807191336-0

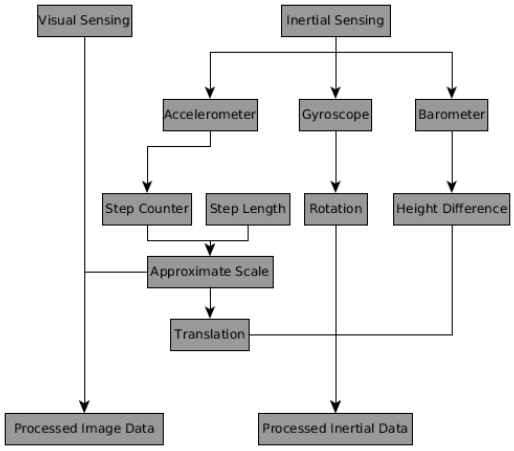

Fig. 1: Detailed step of inertial Sensing

accelerometer z-reading passes the thresholds value including a scaling factor, the threshold value increases to the actual z-reading. This is the advantage of this step counter, because it ensures, that it adopts the walking behavior of the user. In our proposed method, the step also serves as a scale for the visual sensing. Together with the integrated gyroscope readings, DR can be performed. The formula can be seen in (1), where $\vec{x}_i$ is the actual position, $\vec{x}_{i-1}$ is the previous position and R and t are the rotation and translation between both. The principle of the step counter can be seen in Fig. 2.

$$\vec{x}_i = \vec{x}_{i-1} + R * t \tag{1}$$

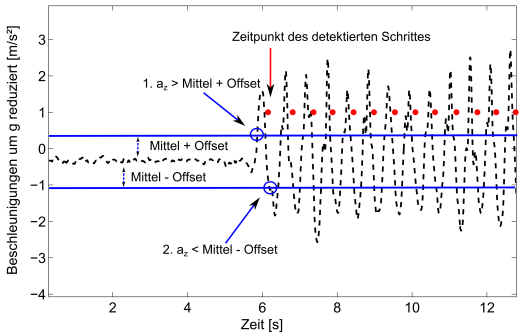

Fig. 2: Principle of the step counter. Black line = accelerometer z-axis reduced by g, blue line = minimal and maximal threshold to detect a step, red dot = detected step [3]

The algorithm in [3] also takes the routing graph with a particle filter into account. The inertial sensing of our method does not require a filter, because the routing graph acts as a line of orientation. If the user navigates near the routing graph and the heading of the smartphone and graph are nearly the same, the position estimated by DR is snapped to the graph and the orientation value is set to the heading of the graph. If the heading is not the same, the algorithm is able to leave the graph, to ensure that the device is able to recognize a trajectory off-sided the graph.

The visual sensing is realized by processing images of the smartphone camera and can be seen in Fig: 3. The principle to get the camera pose is based on feature matching. To ensure that the data obtained from the visual sensing and the inertial sensing are synchronized, the step counter serves as a trigger. Additionally this has the effect, that the translation of the IMU and the visual sensing are equal and the step length can be taken as a scale for the image processing, so there is no need of geo-referencing any anchors.

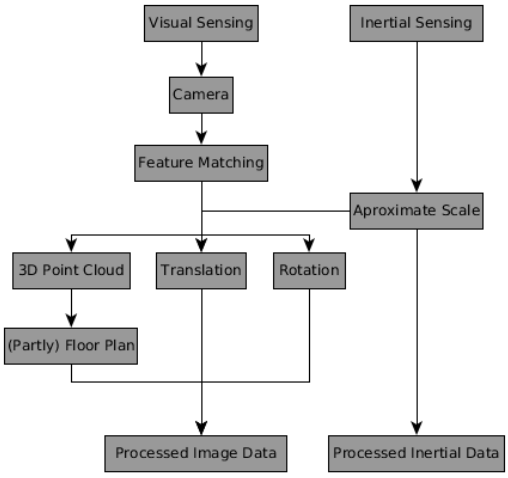

Fig. 3: Detailed step of visual Sensing

Furthermore, the image data is used to obtain 3D point clouds, from which a partly floor plan can be derived. As this system has the aim to serve as an indoor navigation system, there is a given map of the surrounding environment. Both maps can be compared and the scale of the visual translation and the pose of the smartphone can be corrected. This is achieved by a best fit transformation. The pose of the DR can be taken as an approximation for the area of interest, so the computation resources are not used to capacity.

In the current stage of development the visual sensing is realized in post processing, where the images are processed with SURF to find correspondences, which are used to estimate the fundamental matrix and epipolar inliers. With these and the calibration values of the camera a relative camera pose can be detected. To bring the concept onto the smartphone the open source library OpenCV will be used.

Both poses of the modules are going to be fused with a simple kalman filter, to get a more reliable and robust solution. It is assumed that the orientation of the camera is more accurate than the orientation of the IMU, but the translation relies on a scale which is set to the step length. Because of this, the kalman filter weights the orientation of the camera higher than the IMU and the translation are weighted equally.

https://doi.org/10.24355/dbbs.084-201807191336-0

B. Used Hardware

A Samsung Galaxy S8 is used to realize this concept. The specifications are shown in Table I. The phones IMU components range can be set in different areas, but it is assumed that the lowest area ($\pm$ 2 g & $\pm$ 125 $\frac{°}{s}$) are sufficient, as a normal walking behavior should not exceed these values. This smartphone is a good choice because it represents the most spread kind of phone, regarding the camera setup. There are other options like phones that have more cameras to create a depth field, but most of the manufacturers do not provide a library to implement the depth cameras into an own application.

TABLE I: Samsung Galaxy S8 - Specifications [4]

IMU - STMicroelectronics LSM6DSL	
Acc. range	$\pm$ 2 g .. $\pm$ 16 g
Acc. sensivity	0.061 .. 0.488
Angular rate range	$\pm$ 125 $\frac{°}{s}$.. $\pm$ 2000 $\frac{°}{s}$
Angular rate sensivity	4.375 .. 70
Camera - Sony IMX333	
Sensor size	1/2.55 ”
Pixel size	1.4 µm
Resolution	up to 12 MP
FOV	77°
Aperture	F1.7

IV. Experimental Results

A. Inertial Sensing

A test route has been followed and can be seen in Fig. 4. The trajectory starts and ends at the same position. The smartphone was held in a 45° angle in relation to the ground. Table II shows the position estimation of the inertial sensing and has a closure error of 1.83 m. The algorithm recognized 412 steps. These results show that the step counter and the support through the routing graph as an update for position and orientation work quite well.

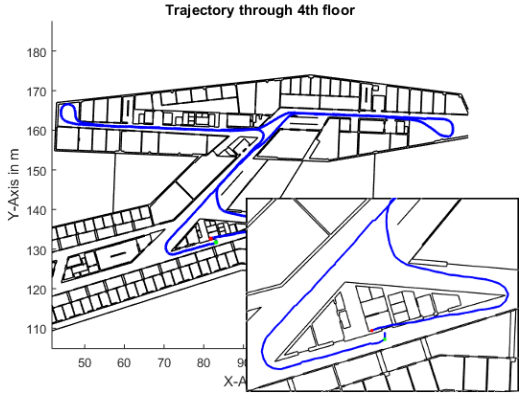

Fig. 4: Trajectory through 4th floor of the HCU building

TABLE II: Results of inertial trajectory

	X [m]	Y [m]
Start	83.100	131.700
End	81.450	132.500
Abs. difference	1.650	0.800
Dist. of closure error	1.830	

B. Visual Sensing

The investigations to prove the visual sensing concept were done in post processing as this module is not implemented on the smartphone yet. They should demonstrate which results are achievable when utilizing smartphone cameras.

1) Trajectory: The visual sensing was tested in its capability to follow the pose along a straight line. The phone was fastened on the sledge of a comparator track and was moved in equal distances along the track. These distances were 0.75 m, as this is assumed by [5] to be the step length for the most people. The step is taken as scale for the processing. The smartphone camera was used to capture a video stream with 1280x720 px. At each position to be processed a sound indicator indicates the time which frame should be taken. The images were processed with the SURF algorithm to get the translation and rotation of the camera, which were used to perform DR.
The results are shown in Table III. It should be noticed that the first position is 0.30 m. The means of the values show that the visual sensing is capable to follow a straight line with an error less than 2 cm.
The processing was repeated 20 times to verify the results are

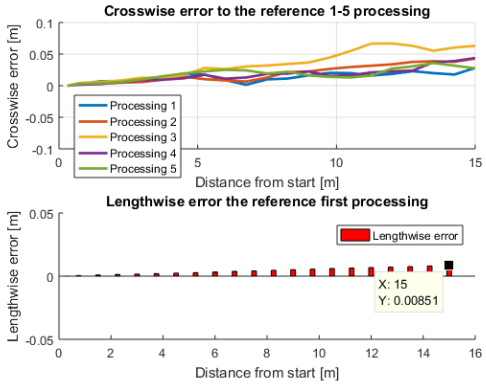

Fig. 5: Length- and crosswise error along a reference

equal. As seen in Fig. 5, where the crosswise error of the first five processing repetitions are pictured, the errors show similar behavior. They drift to the left. It seems that the feature points to the left are weighted more than to the right. This is because the right side to the comparator track is a gray wall with low contrast, and the right side has many different objects in it. They are not exactly equal, because the different processing repetitions have not found the same features. The lengthwise

https://doi.org/10.24355/dbbs.084-201807191336-0

error shows a smaller value and it has a constant inclination. This is because the small errors are accumulated.

TABLE III: Results of visual made trajectory compared to a straight line

Distance [m]	δx [m]	δy [m]	δz [m]	δ [m]
0.30	0.000	0.000	0.000	0.000
0.75	0.003	0.002	0.000	0.004
1.50	0.007	0.002	0.001	0.007
...	...	...	...	...
13.50	0.020	-0.008	0.007	0.023
14.25	0.018	-0.010	0.008	0.022
15.00	0.028	-0.007	0.008	0.030
Mean	0.001	0.005	0.005	0.012

2) Point cloud: The same images were used to derive point clouds from them. A reference point cloud was received by a laserscan. The images were processed with the software PhotoScan from Agisoft. As seen in Fig. 6 gaps in the point cloud occurred due to the fastening of the phone in portrait mode. This is why more of the ceiling was captured than the wall next to the comparator track. Fig. 6 also shows the differences between the reference and the derived point cloud. Points with a low difference are colored blue, while points with a difference up to 0.15cm are colored red.
Fig. 7 shows the allocation of the points into eight classes. The

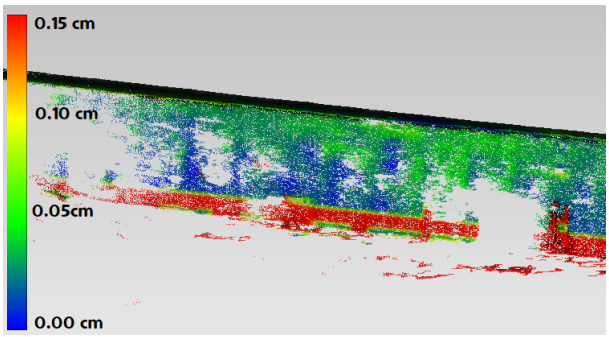

Fig. 6: Differences between the point clouds from images and from laserscanning

intervals of the classes and the number of points as percentage are shown in Table IV. More than 60 % of the points have a lower difference to the reference than 5.6 cm which is a suitable result, regarding to the unstable camera parameters of smartphones.

V. Conclusion and Outlook

This paper presents a concept for a visual inertial indoor navigation system. This concept is based on a smartphone IMU and camera. In addition to the achieved translation and rotation trough the camera, the images can be used to derive 3D point clouds. These can be used to get partly floor plans, which can be compared to a given map to achieve pose and scaling corrections.
First tests show, that each module on its own is able to achieve good results. Especially for the visual sensing more

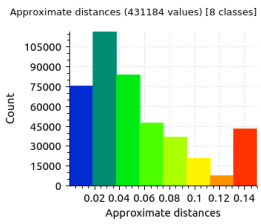

Fig. 7: Histogram of the differences

TABLE IV: Statistics of the histogram

Class	Start [m]	End[m]	[%]
1	0.000	0.019	17.490
2	0.019	0.038	26.999
3	0.038	0.056	19.433
4	0.056	0.075	10.992
5	0.075	0.094	8.517
6	0.094	0.113	4.826
7	0.113	0.131	1.750
8	0.131	0.150	9.993

tests with longer distances and with rotations have to be done. The executed tests are also under optimal conditions, with no rolling shutter effects, which are assumed to be seen in real life conditions. The visual sensing has to be tested while walking with the device in the hand.
While the inertial sensing module is already implemented on a smartphone, the visual sensing is not. The next steps are to implement the visual sensing in a way that both modules can run independently on a smartphone. Furthermore both module data should be combined with a kalman filter to eliminate the weaknesses of each module.
As mentioned before, there are other camera setups available in smartphones and a very interesting device is the Lenovo Phab 2 Pro, as it has an infrared projector available and allows direct measurements of depths. In this way the scaling problem can be solved.

References

[1] L. Atzori, T. Dessi, and V. Popescu, "Indoor navigation system using image and sensor data processing on a smartphone," in *Optimization of Electrical and Electronic Equipment (OPTIM), 2012 13th International Conference on*. IEEE, 2012, pp. 1158–1163.

[2] M. Piras, A. Lingua, P. Dabove, and I. Aicardi, "Indoor navigation using smartphone technology: A future challenge or an actual possibility?" in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*. IEEE, 2014, pp. 1343–1352.

[3] T. Willemsen, "Fusionsalgorithmus zur autonomen positionsschätzung im gebäude, basierend auf mems-inertialsensoren im smartphone," Ph.D. dissertation, HafenCity University Hamburg, 2016.

[4] Samsung. (2018) Specifications. [Online]. Available: https://www.samsung.com/global/galaxy/galaxy-s8/specs/ [Accessed: 28- Jul- 2018]

[5] J. Kupke, T. Willemsen, F. Keller, and H. Sternberg, "Development of a step counter based on artificial neural networks," *Journal of Location Based Services*, vol. 10, no. 3, pp. 161–177, 2016.

https://doi.org/10.24355/dbbs.084-201807191336-0